DOCKET No.

NAI1P002/00.056.01

U.S. PATENT APPLICATION

FOR

SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR DYNAMIC SYSTEM ADAPTATION USING CONTRACTS

INVENTORS: Richard J. Feiertag
Lee Benzinger
Jaisook Rho

ASSIGNEE: NETWORK ASSOCIATES, INC.

KEVIN J. ZILKA

PATENT AGENT

P.O. BOX 721030

SAN JOSE, CA 95172

SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR DYNAMIC SYSTEM ADAPTATION USING CONTRACTS

Richard J. Feiertag
Lee Benzinger
Jaisook Rho

GOVERNMENT LICENSE RIGHTS

The present invention was made with Government support under contract #F30602-97-C-0187 awarded by USAF, AFMC, Rome Laboratory, Directorate of Contracting/PKRZ, 26 Electronic Parkway, Rome, NY 13441-4514. The Government has certain rights in the invention.

FIELD OF THE INVENTION

The present invention relates to networks, and more particularly to dynamically adapting network operation.

BACKGROUND OF THE INVENTION

Systems are very dynamic in that users of the system and devices attached thereto often change on a regular basis. Further, software executed on the system often gets modified and/or updated, and loads on the system periodically fluctuate. There is thus a well established problem of adapting to the ever changing needs of a system.

System adaptation may be required for a number of reasons, both innocent and ill-willed. For example, adaptation may be necessary due to a system being

under attack by a hacker, changing system policy, fluctuating system resource utilization, varying availability of different system resources, and/or changing system performance constraints. Preferably, the approach to adaptation is automated as much as possible so as to effect fluid operation as the system adapts to changing conditions.

In some cases, such automation is handled by algorithms used for scheduling the adaptation of various computer resources. Most resource scheduling algorithms are heuristic, i.e. they are based on rules of thumb. This technique generally works well, but often performs poorly when conditions get out of an expected range, or when the system changes unexpectedly. Unfortunately, modern systems change quite often. For example, in a computer network, a set of computer systems communicate with each other via a high speed connection, and the makeup of the system may be in a constant state of flux, as individual computer systems are connected to and disconnected from the network system, and interoperate in various fashions.

In addition to the changes in the computer system configuration, there is also the issue of a constantly changing system workload. The system workload is the combination of all of the work which various components of the system are processing. This system workload changes periodically with the time of day, day of week, and even the time of year, and may also change permanently over time. It is extremely difficult to derive a set of rules or heuristics for resource allocation which anticipates all of these variations in operating conditions.

Changes in a system are thus unpredictable and may occur unexpectedly in real-time. In order to effectively initiate and achieve adaptation, human intervention is often required. As the system changes, a human operator is often relied upon to assess the status of the system, and implement various algorithms or the like for adapting the system to meet the needs of the various components thereof. Unfortunately, this is very cumbersome and subject to error, resulting in a failure to

meet the requirements of the various components of the system or system requirements for performance, resource utilization, service availability or security. There is thus a need for a more dynamic, automated method of system adaptation.

5

DISCLOSURE OF THE INVENTION

A system, method and computer program product are provided for dynamic adaptation of a system in accordance with a contract with criteria associated therewith. During operation, an interaction between a plurality of components of a system is governed utilizing the criteria of the contract. Further, it is determined whether the interaction between the components of the system meets the criteria of the contract. Upon the criteria of the contract not being met, the interaction between the components of the system is adapted to conform to the criteria.

In a preferred embodiment, the interaction between the components of the system is adapted by adjusting the contract. As an option, the contract may be adjusted using numerous methods. For example, such methods may include deactivation of the contract, modification of the contract, deletion of the contract, and/or activation of a different contract.

In another preferred embodiment, the criteria of the contract may include cost model criteria. Such cost model criteria may be based on resource utilization, performance, service provisioning, etc. Further, the interaction that is governed and adapted among the components may be security-related interaction, performance-related interaction, and/or interaction relative to any other aspect of systems that affects the cost model criteria.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure **1** shows an exemplary system with a plurality of components in accordance with a preferred embodiment;

Figure **2** shows a representative hardware environment for the various components of Figure **1**;

Figure **3** shows steps taken for dynamic adaptation of a system in accordance with a contract having criteria associated therewith;

Figure **4** shows one example of creating a contract between the components of the system;

Figure **5** shows the manner in which the interaction between the components of the system is adapted to conform to the criteria; and

Figure **6** shows an exemplary implementation of a preferred embodiment.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

A dynamic, automated method is provided for adapting an interaction between various components of a system in order to continuously meet system and component requirements in real-time or near real-time. In order to accomplish this, the interaction between components is governed by contracts. These contracts utilize criteria, e.g. cost criteria, to determine that the component interaction meets the contract. During operation, the interaction between the components of the system is automatically adapted to meet the criteria of the contracts.

Figure **1** illustrates an exemplary system **100** with a plurality of components **102** in accordance with a preferred embodiment. As shown, such components include a network **104** which take any form including, but not limited to a local area network, a wide area network such as the Internet, etc. Coupled to the network **104** is a plurality of computers which may take the form of desktop computers **106**, laptop computers **108**, hand-held computers **110**, or any other type of computing hardware/software **111**. As an option, the various computers may be connected to the network **104** by way of a server **112** which may be equipped with a firewall for security purposes. It should be noted that any other type of hardware or software may be included in the system and be considered a component **102** thereof.

A representative hardware environment associated with the various components of Figure **1** is depicted in Figure **2**. In the present description, the various sub-components of each of the components **102** may also be considered components **102** of the system **100.** For example, particular software modules executed on any component **102** of the system **100** may also be considered components **102** of the system **100.** Figure **2** illustrates a typical hardware configuration of a workstation in accordance with a preferred embodiment having a central processing unit **210**, such as a microprocessor, and a number of other units

interconnected via a system bus **212**. The workstation shown in Figure **2** includes a Random Access Memory (RAM) **214**, Read Only Memory (ROM) **216**, an I/O adapter **218** for connecting peripheral devices such as disk storage units **220** to the bus **212**, a user interface adapter **222** for connecting a keyboard **224**, a mouse **226**, a speaker **228**, a microphone **232**, and/or other user interface devices such as a touch screen (not shown) to the bus **212**, communication adapter **234** for connecting the workstation to a communication network **235** (e.g., a data processing network) and a display adapter **236** for connecting the bus **212** to a display device **238**.

The workstation may have resident thereon an operating system such as the Microsoft Windows NT or Windows/95 Operating System (OS), the IBM OS/2 operating system, the MAC OS, or UNIX operating system. It will be appreciated that a preferred embodiment may also be implemented on platforms and operating systems other than those mentioned. A preferred embodiment may be written using JAVA, C, and/or C++ language, or other programming languages along with an object oriented programming methodology. Object oriented programming (OOP) has become increasingly used to develop complex applications.

Figure **3** is a flowchart illustrating a method **300** for dynamic adaptation of a system in accordance with a contract having criteria associated therewith. Such criteria of the contract may be based on various models. For example, the criteria of the contract may be based on a cost model. The cost model criteria may relate to resource utilization. In just one example, such cost model criteria may represent some percentage of resource utilization, or an amount of load on the system. In yet another embodiment, the cost model criteria may relate to performance, i.e. performance degradation. For example, such criteria may include a minimal speed parameter value. Still yet another example of cost model criteria may include provisioning of service. It should be understood that the cost model criteria may be based on any parameter representing a system aspect that is of value to the user. In each of the above examples, levels, standards and/or specifications associated with

the criteria may be predetermined and saved for reasons that will soon become apparent.

As an option, a set of contracts may govern the interaction between any plurality of components. In this case, each contract of the set has unique criteria associated therewith. In such embodiment, each contract of the set may include criteria indicating which other contract of the set should be implemented upon a condition being met. In one example, such condition may include the satisfaction and/or failure of the system to meet the contract criteria.

As shown in Figure **3**, it is first determined in decision **301** whether there a contract exists. If not, a contract is created in operation **302**. The initial formation of the contract(s) may be carried out in various ways. For instance, the components or the management entities for the components may be permitted to negotiate the criteria. In real-time or near real-time systems, control entities may be permitted to negotiate the criteria. This negotiation may be accomplished by any desired method one example of which will be set forth in greater detail during reference to Figure **4**.

During operation **303**, an interaction between the components of the system is governed utilizing the criteria of the contract. In particular, the system and components are governed or controlled in a constant effort to meet the criteria of the contract. While the interaction is being governed in operation **303**, it is possible that the interaction may terminate. Conditions for determination of interaction termination **304** may be implemented as part of the conditions of the criteria of the contract. Conditions for determination of interaction termination **304** may also be implemented within the management and control structure for the components. In either case, when a determination is made that in **304** that the components are no longer interacting, the method **300** is ended.

With continuing reference to Figure **3**, it is determined in decision **306** whether the interaction between the components of the system meets the criteria of

the contract. This may be accomplished by monitoring various system parameters from which it can be determined whether the criteria are met. Upon it being determined in decision **306** that the criteria of the contract are not being met, the interaction between the components of the system is automatically adapted to conform to the criteria. Note operation **308**. It should be understood that any one of the components of the system may be designated to carry out such adaptation. Additional detail will be set forth regarding the system adaptation of operation **308** in Figure **5**.

In a preferred embodiment, the interaction between the components of the system is adapted by adjusting the contract, thus altering the criteria. As an option, the contract may be adjusted using numerous methods. For example, such methods may include deactivation of the contract, modification of the contract, deletion of the contract, and/or activation of a different contract. After system adaptation, the interaction between the components of the system is governed utilizing the new criteria of the adjusted contract, as set forth in operation **303**.

Figure **4** is a flowchart illustrating one example of creating a contract between the components of the system in accordance with operation **302** of Figure **3**. First, in operation **400**, a requesting component of the system advertises its capability to a control component which handles contract negotiations. Thereafter, in operation **402**, the control component sends the requesting component a proposed contract based on the capability and any other pertinent factors. As an option, the control component may choose a contract from a predetermined set. If the requesting component does not accept, the process is terminated and interaction is prohibited.

Upon the requesting component accepting the proposed contract, the control module sends a seal message to the requesting component thereby completing the contract negotiation. Note operation **404**. As an option, the control module may also generate a fallback proposed contract and send the same to the requesting

component, as indicated in operation **406**. Upon receipt, the requesting component verifies the fallback proposed contract in operation **408**. Again, in operation **410**, the control module sends a seal message completing the fallback contract negotiation. It should be noted that the fallback contract is stored, but not activated, for use in case the first contract requires a replacement. This replacement can occur when the interaction governed by the first contract no longer meets the criteria of the cost model.

Another exemplary contract(s) negotiation scheme that is derived from the principles set forth herein may be found with reference to "Intrusion Detection Inter-Component Adaptive Negotiation" by Lee Benzinger et al. which is incorporated herein by reference in its entirety. See Appendix A.

Figure **5** is a flowchart illustrating the manner in which the interaction between the components of the system is adapted to conform to the criteria in accordance with operation **308** of Figure **3**. As shown, it is first determined how the system is to adjust the contract to achieve system adaptation. Note decision **500**. This may be indicated within the contract under which components are currently operating or by another desired means. Based on the results of decision **500**, a preferred embodiment may be capable of performing various functions.

In operation **502**, the contract may be adjusted by simply deactivating the contract. In the alternative, the contract may be deleted in operation **504**. Of course, the deactivation or deletion of the contract would not provide any means of continuing interaction, thus terminating the same. Still yet, the contract may be modified in operation **506**. Such may include modification of the criteria by a predetermined formula, negotiation between parties, or any other desired means. As yet another option, a different contract, i.e. the fallback contract, may be executed in operation **508**. Selection of the existing contract may be accomplished by simply selecting a separate contract stored in a database.

Figure **6** is a schematic diagram illustrating an exemplary implementation of a preferred embodiment. Such example is set forth in the context of handling security-related events in a system via interaction and adaptation between the various components. In particular, the present example features how the system of a preferred embodiment may react to an intrusion by a hacker. It should be noted, however, that a preferred embodiment may be implemented to interact and adapt in response to any type of event, or even general aspects of the system.

As shown in Figure **6**, a plurality of components is interconnected in a manner similar to that shown in Figure **1**. In particular, a lap-top computer **600** is coupled to a local area network (LAN) **602** via the Internet **604**. A firewall **606** is coupled between the Internet **604** and the LAN **602**. The LAN **602** includes a pair of intrusion detection modules **608** and an analysis module **610**. Prior to use, a set of contracts is established for each of the components of the LAN **602,** the firewall **606** and the analysis module **610**.

During use, such contracts dictate the manner in which the components of the LAN **602** and the firewall **606** interact and adapt to changing network conditions, as set forth earlier during reference to Figure **3**. In particular, the intrusion detection modules **608** are capable of communicating information being sent to and received from components behind and beyond the firewall **606** for the purposes of detecting intrusions and anomalies in the system.

In order to communicate information to and from other components in the system, the intrusion detection modules **608** and any other system component periodically send and receive generalized intrusion detection objects (GIDOS) which are data structures with information that may include, but not be limited to IP addresses of sources of the information, a timestamp indicating when the information was generated, an identifier of a host machine from which the information is received, a descriptor (an attack code) of a particular intrusion on the system that has been detected, and/or a user name of the entity which sent or

received information. It should be understood that any type of data structure may be employed in lieu of GIDOS. For more general information on GIDOS, protocols, and other concepts derived from the principles set forth herein, reference should be made to "Gido Filtering Requirements and Design" by Stuart Staniford-Chen, and "IDIAN Protocol Design Version 1" by Lee Benzinger et al. which are incorporated herein by reference in their entirety. See Appendices B and C, respectively

The GIDOS are sent back and forth between the intrusion detection modules **608** and any other components through the analysis module **610.** The analysis module **610** also has access to the contracts associated with each of the components of the system. Given the information supplied by the GIDOS, the analysis module **610** may thus govern the interaction between the intrusion detection modules **608** and any other components in order to ensure that the criteria of the contracts are met. If they are not, the analysis module **610** may adapt the interaction by adjusting the contracts in a manner set forth hereinabove during reference to Figure **3**.

In the present scenario, the lap-top computer **600** is employed by a traveling user to interact with the various components of the LAN **602.** Further, the lap-top computer **600** is compromised by a hacker who wishes to infiltrate the system. Initially, the lap-top computer **600** may communicate with the analysis module **610** in order to establish, i.e. negotiate, etc., a contract(s) dictating criteria for interaction therewith.

In order to infiltrate the system, a distributed denial of service attack is executed by the lap-top computer **600**. Such denial of service attack may include any method that overuses the resources of the system, thus compromising operability. In the present example, this is accomplished by overloading the intrusion detection modules **608** with GIDOS.

In response to the large influx of GIDOS, the analysis module **610** monitors whether it violates the criteria. In the case where the analysis module **610** utilizes a

contract having cost model-based criteria, a predetermined maximum load amount of GIDOS/sec may be used as a trigger point to initiate system adaptation. It should be appreciated that such maximum load amount may be associated with the system as a whole, or with each of the components of the system individually. In order to determine whether or not the maximum load amount has been reached, the various components of the system are adapted to communicate current load amounts, in addition to any other necessary parameters, to the analysis module **610** using GIDOS. Of course, the maximum load amounts are set to ensure that enough system operability is maintained in order to allow communication of GIDOS for adaptation purposes.

Assuming that such maximum amount has been reached, the analysis module **610** may adapt the system per the terms of the contract. For example, if the GIDOS identify the lap-top computer **600** as performing a suspect attack, the contract associated with the lap-top computer **600** may be adjusted to restrict access to system resources. In a preferred embodiment, a fallback contract may be used which was negotiated before the lap-top computer **600** was permitted to use the system. In cases where there is a plurality of components such as the lap-top computer **600**, the analysis module **610** may optionally adjust contracts of all of the components of the system in order to restrict access to system resources.

If after the system operates under the adjusted contract the system stabilizes to an acceptable level, the fallback contract may dictate that the original contract be reactivated in place of the fallback contract. If this results in a situation where the system is again under attack, the contract may then indicate that the lap-top computer **600** is most likely performing a forbidden attack, and the firewall **606** may be used to prevent any further communication with the Internet **604**. In the alternative, the contract associated with the lap-top computer **600** may be deactivated or deleted.

The foregoing example is set forth in the context of handling security-related events by allowing such events to operate as a trigger for system adaptation. Additional details on the above example which were derived from the principles set forth herein may be found with reference to "Intrusion Detection Inter-Component Adaptive Negotiation" by Lee Benzinger et al. which is incorporated herein by reference in its entirety. See Appendix A. It should be noted, however, that a preferred embodiment may be implemented to interact and adapt in response to any type of event, or in response to general system conditions rather than any particular event.

While various embodiments have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of a preferred embodiment should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

Add Appendixes